Decision Problem: problem with YES/NO answers

SAT: Given φ, does φ have a satisfying assignment?

VC: Given G, k, does G have a vertex cover S such that $|S| \leq k$?

Search Problems:

FIND-SAT: Given Boolean formula φ, return a satisfying assignment (or NIL if none)

FIND-VERTEX-COVER: Given undirected G, return a minimum-size vertex cover

Decision Problem: problem with YES/NO answers

SAT: Given φ, does φ have a satisfying assignment?

VC: Given G, k, does G have a vertex cover S such that $|S| \leq k$?

Search Problems:

FIND-SAT: Given Boolean formula φ, return a satisfying assignment (or NIL if none)

FIND-VERTEX-COVER: Given undirected G, return a minimum-size vertex cover

$SAT \in P \iff FIND\text{-}SAT \in P$

$\Leftarrow$: SAT(φ)
return FIND-SAT(φ) $\neq$ NIL

$SAT \leq_P FIND\text{-}SAT$

$VC \in P \iff FIND\text{-}VC \in P$

$\Leftarrow$: VC(G, k)
$C \leftarrow$ FIND-VC(G)
return $|C| \leq k$

$VC \leq_P FIND\text{-}VC$

Decision Problem: problem with YES/NO answers

SAT: Given φ, does φ have a satisfying assignment?

VC: Given G, k, does G have a vertex cover S such that $|S| \leq k$?

Search Problems:

FIND-SAT: Given Boolean formula φ, return a satisfying assignment (or NIL if none)

FIND-VERTEX-COVER: Given undirected G, return a minimum-size vertex cover

$SAT \in P \iff FIND\text{-}SAT \in P$

$\Leftarrow$: SAT(φ)
return FIND-SAT(φ) $\neq$ NIL

$SAT \leq_P FIND\text{-}SAT$

$VC \in P \iff FIND\text{-}VC \in P$

$\Leftarrow$: VC(G, k)
$C \leftarrow$ FIND-VC(G)
return $|C| \leq k$

$VC \leq_P FIND\text{-}VC$

SAT $\in$ P $\iff$ FIND-SAT $\in$ P

SEARCH $\leq_p$ DECISION

$\rightarrow$: We show FIND-SAT $\leq_p$ SAT (SAT is self-reducible)

FIND-SAT(φ) ← assume variables are $x_1, \ldots, x_n$

```
if SAT(φ) = NO then return NIL

φ' ← φ
A ← [NIL, ..., NIL]
for i = 1 to n
    φ_T = φ' with x_i = T
    φ_F = φ' with x_i = F
    if SAT(φ_T)
        φ' ← φ_T
        A[i] = T
    else
        φ' ← φ_F
        A[i] = F
return A
```

$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee (\neg x_2 \wedge \neg x_3))$

$\varphi' = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee (\neg x_2 \wedge \neg x_3))$

$A \leftarrow [NIL, NIL, NIL]$

$\varphi_T \leftarrow (T \vee x_2 \vee x_3) \wedge (F \vee (\neg x_2 \wedge \neg x_3))$

$\varphi_F \leftarrow (F \vee x_2 \vee x_3) \wedge (T \vee (\neg x_2 \wedge \neg x_3))$

(can simplify)

$A \leftarrow [T, NIL, NIL]$

SAT $\in$ P $\iff$ FIND-SAT $\in$ P

SEARCH $\leq_p$ DECISION

$\rightarrow$: We show FIND-SAT $\leq_p$ SAT (SAT is self-reducible)

FIND-SAT(φ) ← assume variables are $x_1, \ldots, x_n$

if SAT(φ) = NO then return NIL

$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\sim x_1 \vee (\sim x_2 \wedge \sim x_3))$

$\varphi' \leftarrow \varphi$

$A \leftarrow [NIL, \ldots, NIL]$

for i = 1 to n (n iterations)

$\varphi_T = \varphi'$ with $x_i = T$] poly

$\varphi_F = \varphi'$ with $x_i = F$

if SAT(φ_T) (n calls to SAT in loop (+1 above)) poly

$\varphi' \leftarrow \varphi_T$

A[i] = T

else

$\varphi' \leftarrow \varphi_F$

A[i] = F

return A

$\varphi' = (x_1 \vee x_2 \vee x_3) \wedge (\sim x_1 \vee (\sim x_2 \wedge \sim x_3))$

$A \leftarrow [NIL, NIL, NIL]$

$\varphi_T \leftarrow (T \vee x_2 \vee x_3) \wedge (F \vee (\sim x_2 \wedge \sim x_3))$

$\varphi_F \leftarrow (F \vee x_2 \vee x_3) \wedge (T \vee (\sim x_2 \wedge \sim x_3))$

(can simplify)

$A \leftarrow [T, NIL, NIL]$

INVARIANT: a) φ' is satisfiable

b) A + satisfying assignment for φ' satisfies φ

c) A has assignments for $x_1, \ldots, x_{i-1}$

SAT $\in$ P $\iff$ FIND-SAT $\in$ P

SEARCH $\leq_p$ DECISION

$\rightarrow$: We show FIND-SAT $\leq_p$ SAT (SAT is self-reducible)

FIND-SAT(φ) ← assume variables are $x_1, \ldots, x_n$

$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee (\neg x_2 \wedge \neg x_3))$

```
if SAT(φ) = NO then return NIL
φ' ← φ
A ← [NIL, ..., NIL]
for i = 1 to n
    φ_T = φ' with x_i = T
    φ_F = φ' with x_i = F
    if SAT(φ_T)
        φ' ← φ_T
        A[i] = T
    else
        φ' ← φ_F
        A[i] = F
return A
```

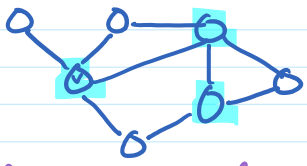
$\varphi' = (T \vee x_2 \vee x_3) \wedge (F \vee (\neg x_2 \wedge \neg x_3))$ $i = 2$

What are φ_T and φ_F for the $i=2$ iteration?

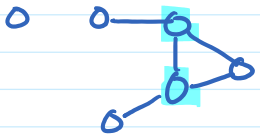
Which one is satisfiable?

FIND-VERTEX-COVER $\leq_P$ VERTEX-COVER:

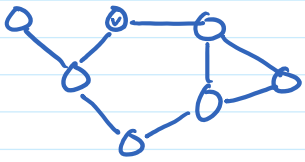
```
FIND-VERTEX-COVER(G)
    k ← n
    while k > 0 and VERTEX-COVER(G, k-1)        ← n calls to VERTEX-COVER
        k ← k-1

poly    C ← ∅
        G' ← G
        for each vertex v
poly        G'' ← G' with v and incident edges removed
            if VERTEX-COVER(G'', k-1)           ← n calls to VERTEX-COVER
poly            G' ← G''
                C ← C ∪ {v}
                k ← k-1
    return C
```



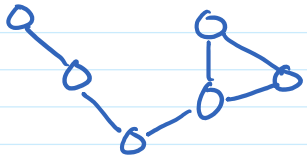
G has vertex cover of size 3 that includes v



G-v has a vertex cover of size 2



G has no vertex cover of size 3 that includes v



G-v has no vertex cover of size 2 (if it did, then that cover + v is a vertex cover of G of size 3)

DECISION EP $\leftrightarrow$ SEARCH EP